

CAPÍTULO 7: IMPLEMENTACIÓN DE LA HERRAMIENTA DE PREDICCIÓN DE COBERTURA

7.1.- Introducción

En la etapa de planificación radioeléctrica contemplada en la metodología para el despliegue de una red Wi-Fi, resulta de gran utilidad el software de cobertura radioeléctrica. Este programa permite el ahorro de numerosos y complicados cálculos, imposibles de realizar a mano, referidos a la obtención del nivel de señal en cada punto del área de despliegue, en función de la ubicación del transmisor y de las características de la zona.

La herramienta de planificación implementada en este Proyecto, desarrollada en su totalidad en el entorno Matlab, permite dimensionar el sistema de comunicaciones para garantizar niveles de cobertura y capacidad adecuados a las necesidades y a la previsión de consumo de los usuarios. El objetivo de esta herramienta es guiar el proceso de diseño del despliegue de la red para asegurar el cumplimiento de los niveles de cobertura y capacidad exigidos por la demanda de servicio prevista, a través de una interfaz intuitiva y fácil de usar.

La aplicación desarrollada para este fin es una sencilla herramienta que permite predecir las coberturas de cualquier punto de acceso en un entorno de trabajo en interiores, así como el diseño de radioenlaces. Su manejo es tan sencillo como el de una instalación guiada de cualquier programa: pantallas correlativas con botones de SIGUIENTE/ATRÁS que te permiten navegar de una forma rápida y eficaz.

La interfaz gráfica que se ha desarrollado en este Proyecto permite al usuario cargar el plano de planta de un área que desea estudiar por lo que se hará uso de los algoritmos explicados en el capítulo 4 para procesar esa imagen y detectar las líneas correspondientes a los muros internos y externos del entorno en interiores.

El método *Ray Tracing 2D* y *3D*, que se ha empleado en esta herramienta, fue elaborado casi en su totalidad, en el proyecto “Modelo de Cobertura para Redes Inalámbricas de Interiores” [2]. Además se le ha añadido el método *Cost 231* para integrarlos en una interfaz gráfica de forma sencilla e intuitiva.

7.2.- Requisitos del Simulador de Cobertura Radioeléctrica para Redes Inalámbricas en Interiores

En este apartado se identifican los requisitos que debe cumplir el modelo propuesto. Estos requisitos afectan básicamente al diseño y a la construcción del mismo.

Se pretende que el modelo se pueda aplicar a entornos reales (recintos existentes) para determinar niveles de cobertura a partir de puntos de acceso dispuestos por el modelo geométrico bidimensional o tridimensional del recinto; o para encontrar la situación óptima de colocación de los puntos de acceso.

Capacidad de importar la geometría del recinto

En el modelo se debe poder importar la geometría de los recintos en los que se quiere estudiar la cobertura inalámbrica, que se hayan modelado utilizando herramientas de modelado estándar existentes.

El modelo se basa en datos geométricos del entorno importados, por lo que es imprescindible obtener el modelo del recinto mediante herramientas externas.

Capacidad de importar las propiedades de los materiales del recinto

Además de importar la geometría del recinto, se deben obtener características especiales de los materiales que lo forman (atenuación al ser atravesados, coeficiente de reflexión, etc.), los parámetros adecuados para el modelo de propagación en LOS utilizado, y cualquier otra información necesaria. Estos datos, además de introducirse interactivamente en el prototipo, también podrán ser almacenados y cargados en bloque.

Coste computacional aceptable

El coste computacional de los cálculos necesarios debe ser aceptable en los sistemas actuales. No se trata de obtener un modelo en tiempo real, pero los tiempos de cálculo no deberán exceder de unas pocas horas para entornos reales de un edificio con varias plantas.

Requisitos del prototipo

Para comprobar la validez del modelo, el prototipo que implemente el modelo se pretende que cumpla también una serie de requisitos, que se muestran a continuación.

- Interactivo

Uno de los principales usos que se prevén para el modelo es la ubicación óptima de puntos de acceso. Si los resultados obtenidos no satisfacen las necesidades, o si se desea buscar configuraciones mejores, será preciso realizar pruebas y ensayos modificando la situación de los puntos de acceso, o modificando parámetros del sistema.

El prototipo debe permitir realizar cambios y modificaciones en los datos, que serán tenidos en cuenta inmediatamente para nuevos cálculos, y por tanto, responder a las entradas de los usuarios. Por ejemplo, se deben poder ajustar parámetros de propagación, propiedades de los materiales, los puntos de acceso se deben poder reconfigurar en cuanto a potencia, situación, características de la antena, etc.

- Interfaz gráfica

El sistema debe permitir, en un entorno gráfico, la realización de todas las operaciones: importación de modelos geométricos, ajuste de parámetros, obtención de resultados en diferentes vistas, almacenamiento de modelo completo (modelo tridimensional junto a características de propagación).

- Exportación de datos

Los datos introducidos en el sistema se deben poder almacenar para su posterior carga y realización de nuevos cálculos.

Los resultados de la simulación deben ser exportables en diferentes vistas: imágenes, modelo tridimensional, valores de señales.

- Usabilidad

La definición formal aplicada por la organización estándar internacional ISO (*International Organization for Standardization*) establece que la usabilidad es la “eficacia, eficiencia y satisfacción con la que un conjunto de usuarios pueden afrontar un conjunto de tareas en un entorno particular”.

La definición anterior tiene un significado operacional pero también requiere una cierta medida de eficacia, eficiencia y satisfacción para su desempeño. El concepto de usabilidad es asociado muchas veces a las aplicaciones cuya interfaz de usuario juega un papel importante, sin embargo, es un concepto amplio que debe estar aplicado al desarrollo de cualquier producto software.

El prototipo deberá ser diseñado para que integre todos los aspectos relacionados con la usabilidad. Esta característica permitirá que el sistema sea eficiente, ofrezca un entorno visual agradable, sea fácil de aprender con un nivel de abstracción adecuado para la retención de ideas, sea fácil de usar, incorpore mecanismos para poder tratar y subsanar los errores rápidamente y ofrezca un buen grado de satisfacción por parte del usuario.

- Calibrado del modelo y comprobación de resultados

El prototipo debe poder ser calibrado con medidas realizadas en el entorno, que permitan ajustar los parámetros de propagación al entorno real.

Por otra parte, el prototipo debe tener la capacidad de aceptar medidas de señal reales, para que se puedan comparar con los resultados de señal calculados en los diversos modelos de propagación, y aplicados al entorno modelado.

7.3.- Manual de usuario del simulador

En este apartado se va a detallar el funcionamiento de la herramienta desarrollada en este Proyecto mediante una serie de pasos centrándose en la interfaz de simulación de cobertura radioeléctrica en 2D con el modelo de propagación *Ray Tracing*.

En primer lugar, al ejecutar el fichero simulador.exe aparecerá la pantalla principal o portada.

7.3.1.- Pantalla principal

Desde la pantalla principal que se representa en la figura 7.1, se puede acceder a un menú desplegable (*popupmenu*), para seleccionar entre los tres disponibles, el tipo de modelo de propagación elegido para realizar una simulación:

- *2D Ray Tracing*
- *3D Ray Tracing*
- *COST 213 CMWM*

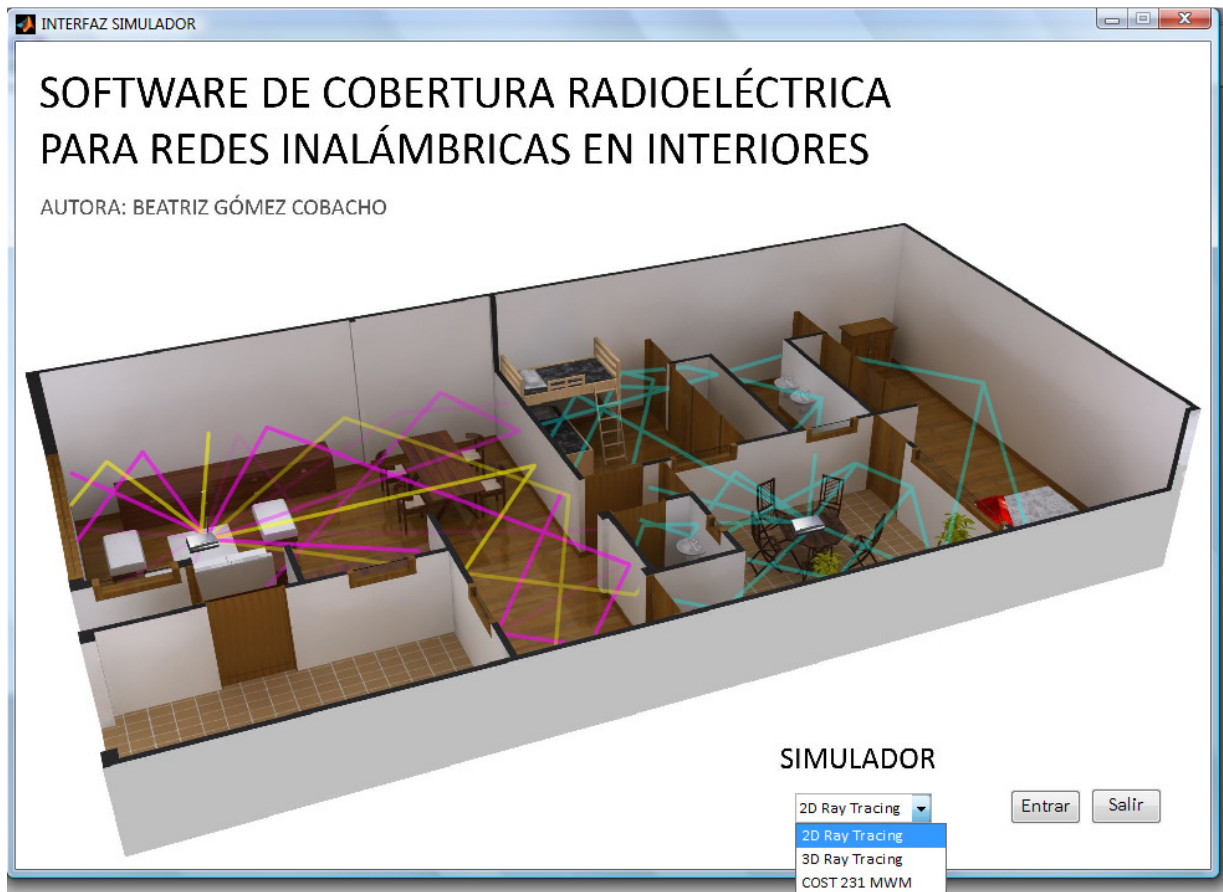


Figura 7.1: Portada del simulador de Cobertura Radioeléctrica

Una vez seleccionado el método que se quiere emplear para el estudio, se pulsa el botón 'Entrar'.

Al ejecutarlo, se despliega una ventana como la que se muestra en la figura 7.2 que permite al usuario introducir parámetros característicos del estudio que realiza, tales como: polarización de las antenas, la potencia del transmisor, resolución, etc.

Estos datos se cargan en variables, para ser utilizadas posteriormente en el programa principal. Estas variables se almacenan en ficheros .mat que se guardan con el comando 'save' y se cargan cuando con el comando 'load' (ver Anexo I).

La directividad de las antenas no es un parámetro que se tendrá en cuenta por el software: siempre se asumen omnidireccionales ideales. Esto es debido a que los sistemas que se pretenden simular son WLAN's y en general utilizan antenas de tipo omnidireccional.

7.3.2.- Interfaz gráfica principal de adquisición de datos

En la figura 7.2 se puede observar la interfaz de adquisición de datos principal. Ésta estará compuesta por:

- 'Axes', donde se mostrará el plano introducido por el usuario antes y después del tratamiento de la imagen.
- 'Edit text', donde se introducirán valores como las dimensiones máximas del entorno a simular (x_max e y_max) o la potencia del trasmisor.
- 'Push button', los cuales ejecutarán una determinada función o rutina al hacer clic sobre ellos o abrirán otras ventanas (*figures*).
- 'Popupmenu', para elegir un valor entre los propuestos en la lista. Se emplearán los *popupmenu* para seleccionar la resolución requerida para la presentación de resultados o para elegir la polarización de la antena.

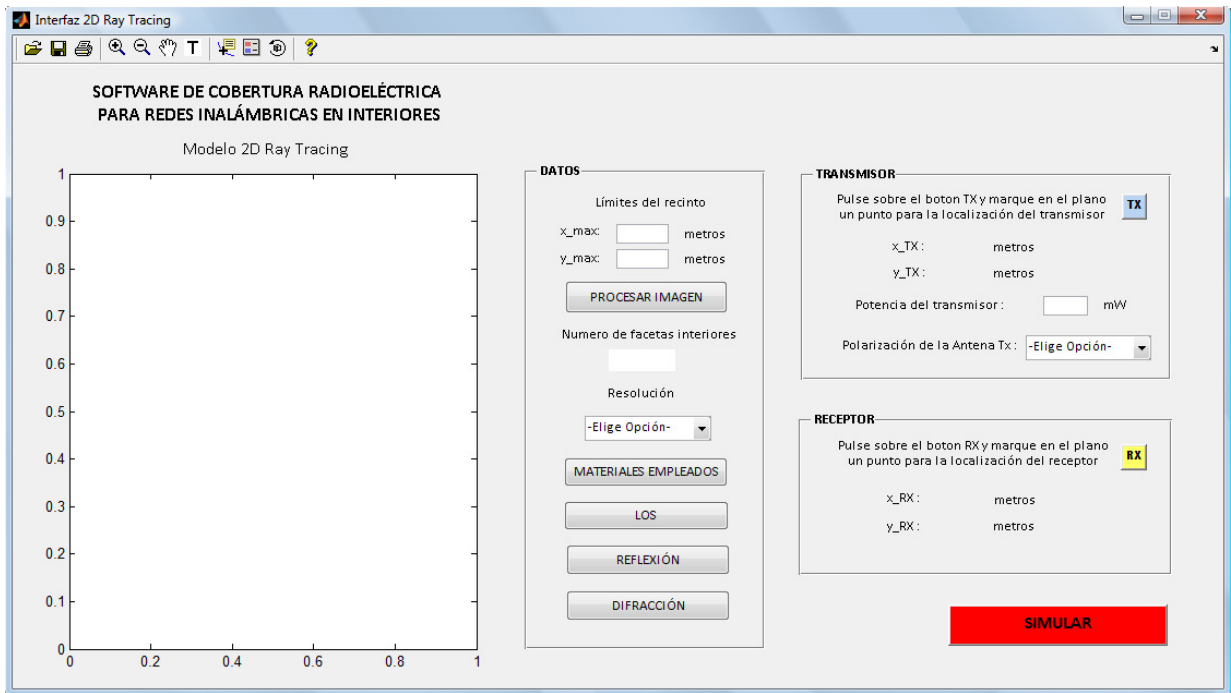


Figura 7.2: Interfaz principal de adquisición de datos


El procedimiento para realizar una simulación en el entorno en interiores deseado se expone en los siguientes apartados.

7.3.2.1.- Cargar Plano

Esta herramienta permite al usuario importar una imagen de un plano de planta sobre el cual desea hacer el estudio. Dicha imagen debe estar en formato *jpeg*, *bmp*, *tiff*, *png* o *gif*. Una opción es tomar el plano de planta de un archivo tipo *CAD*, y pasarlo a un editor de imágenes guardándolo en uno de estos formatos.

El plano de planta se guarda como una imagen indexada en Matlab, en un arreglo tridimensional ($m \times n \times 3$), representando la imagen original, en la que el valor de cada componente corresponde al color de dicha posición, o si se trata de una imagen en escala de grises se guardará en una matriz bidimensional.

Las paredes se representarán con líneas simples continuas, sin huecos ni líneas curvas. No deben estar representadas puertas ni ventanas, solo las líneas de las paredes (Figura 7.4)

En primer lugar, se hace clic con el ratón sobre el icono de Abrir  para insertar la imagen del plano sobre el que se quiere obtener la cobertura radioeléctrica.

Se emplea la función *imread* de Matlab para leer la imagen:

```
[nombre direc]=  
uigetfile({'*.jpg;*.jpeg;*.bmp;*.gif;*.png;*.tiff'}, 'Abrir Imagen');  
  
if nombre==0  
    return  
else  
    img = imread(fullfile(direc,nombre));
```



Figura 7.3: Planta real empleada en los ejemplos del simulador

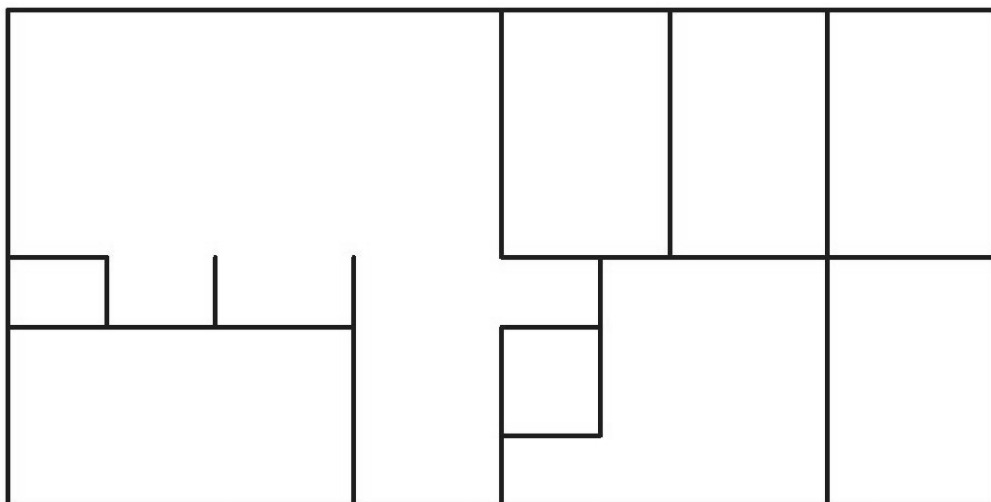


Figura 7.4: Imagen del plano de la planta anterior en formato .jpg

Una vez cargado el plano en el *axes* de la interfaz se introducen los límites de éste con el objetivo de escalarlo y pasar de píxeles a metros.

A continuación se pulsa sobre el botón ‘**PROCESAR IMAGEN**’ (Figura 7.5).



El formulario, titulado "Límites del recinto", contiene dos campos de entrada de texto. El primer campo, etiquetado como "x_max:", contiene el número "8" y está seguido por el texto "metros". El segundo campo, etiquetado como "y_max:", contiene el número "10" y está seguido por el texto "metros". Debajo de estos campos hay un botón rectangular con el texto "PROCESAR IMAGEN".

Figura 7.5: Cuadro para introducir los límites del recinto

El software emplea un algoritmo para reconocer todas las líneas del plano, mediante el que se determinará el número de facetas (paredes) y las coordenadas de cada una de ellas.

A continuación se explicará detalladamente el algoritmo de procesamiento de imagen con el ejemplo concreto del plano que se emplea en los ejemplos de simulación.

7.3.2.1.1.- Implementación del algoritmo de procesamiento de imagen en el simulador

Al hacer clic sobre el Botón ‘Procesar imagen’ se llama a la función *detectalineas.m* que procesa la imagen almacenada en la variable ‘*img*’.

El siguiente código muestra las funciones principales que se han empleado.

```
Igris = rgb2gray(img); % convierte a intensidad
BW = edge(Igris, 'canny', [0.1 0.2]); %Algoritmo de canny para detección
de bordes
[H, T, R] = hough(BW); %Transformada de Hough para detección de líneas
imshow(H, [], 'XData', T, 'YData', R, 'InitialMagnification', 'fit');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
P = houghpeaks(H, 5, 'threshold', ceil(0.1*max(H(:))));
x = T(P(:,2)); y = R(P(:,1));
```

```

plot(x,y,'s','color','white');

% Encuentra líneas y las representa

lines = houghlines(BW,T,R,P,'FillGap',5,'MinLength',40);

xy_max = [0 0; 0 0];

factor_escalax = x_max_plano/xmax_pixel;
factor_escalay = y_max_plano/ymax_pixel;

num_facetas = length(lines);

% Para cada línea detectada se hace un escalado para pasar de pixel a
metros y se representa

for k = 1: length (lines)

xy = [lines(k).point1; lines(k).point2];

% xy pasado a metros
xy_escala = [xy(:,1)*factor_escalax,xy(:,2)*factor_escalay]

plot(xy_escala(:,1),xy_escala(:,2),'LineWidth',2,'Color','green');

% Punto de comienzo de faceta
plot(xy_escala(1,1),xy_escala(1,2),'x','LineWidth',2,'Color','yellow')

% Punto de final de faceta
plot(xy_escala(2,1),xy_escala(2,2),'x','LineWidth',2,'Color','red');

end

```

De la figura 7.6 a la 7.9 se puede ver el proceso que experimenta la imagen desde que se introduce en la interfaz hasta que se reconocen las líneas. En las figuras 7.10 y 7.11 se ve la interfaz 2D y 3D una vez procesada la imagen y reconocidas las líneas del plano.

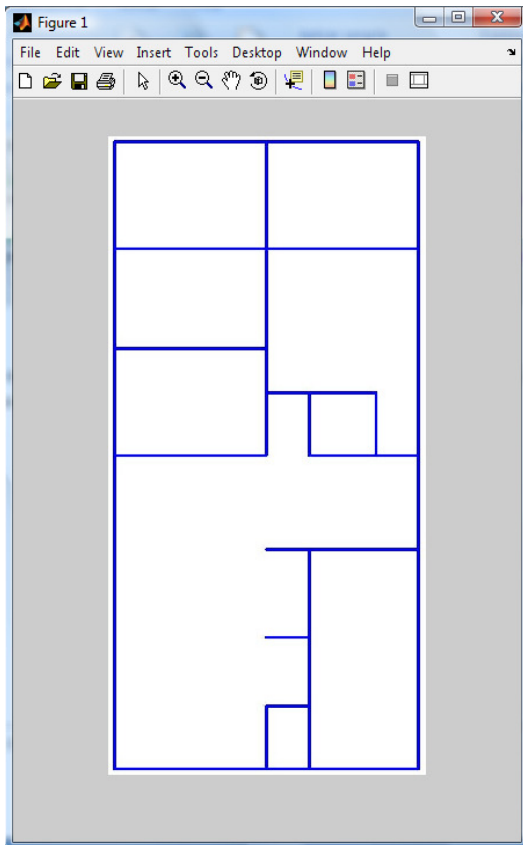


Figura 7.6: Imagen original RGB (Plano)

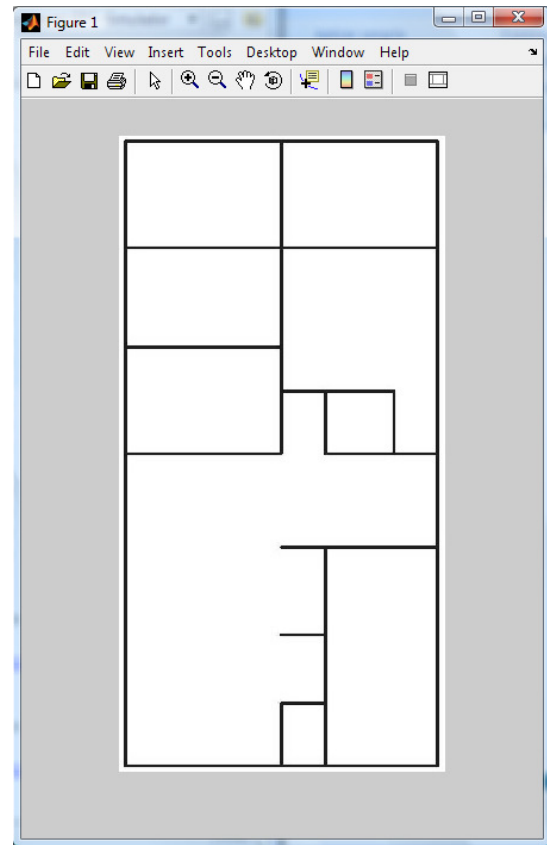


Figura 7.7: Imagen en escala de grises

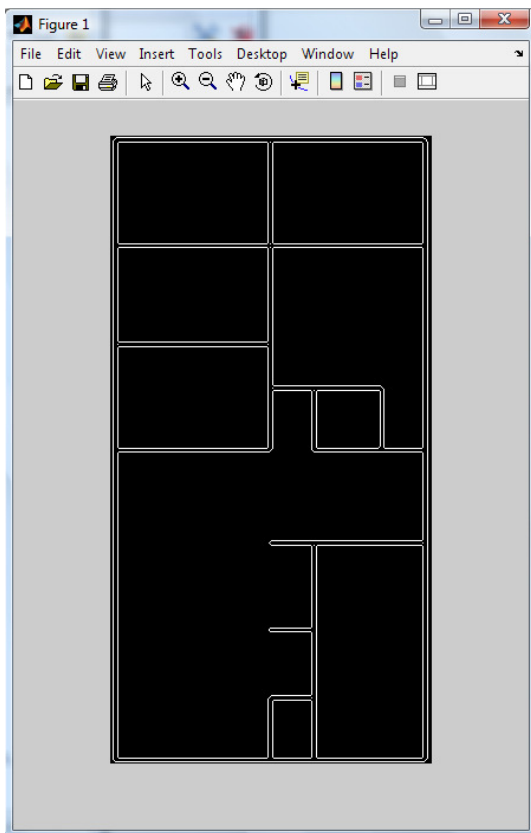


Figura 7.8: Imagen Binaria. Algoritmo de Canny (detección de bordes)

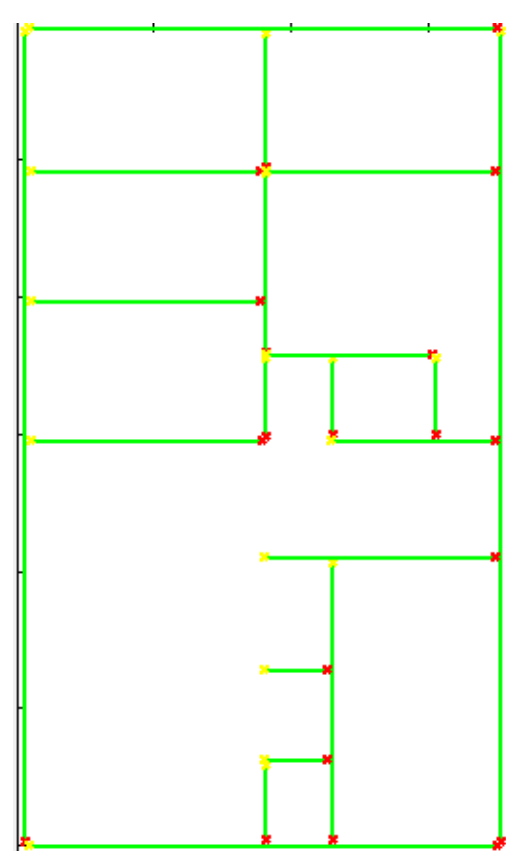


Figura 7.9: Transformada de Hough (detección de líneas)

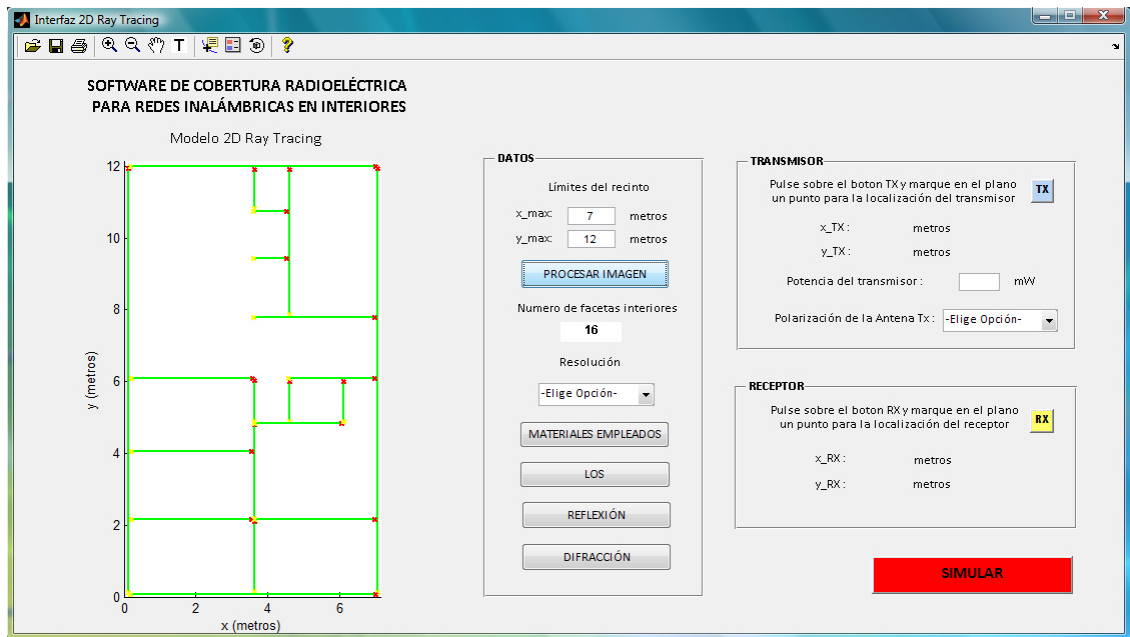


Figura 7.10: Interfaz 2D después de procesar plano

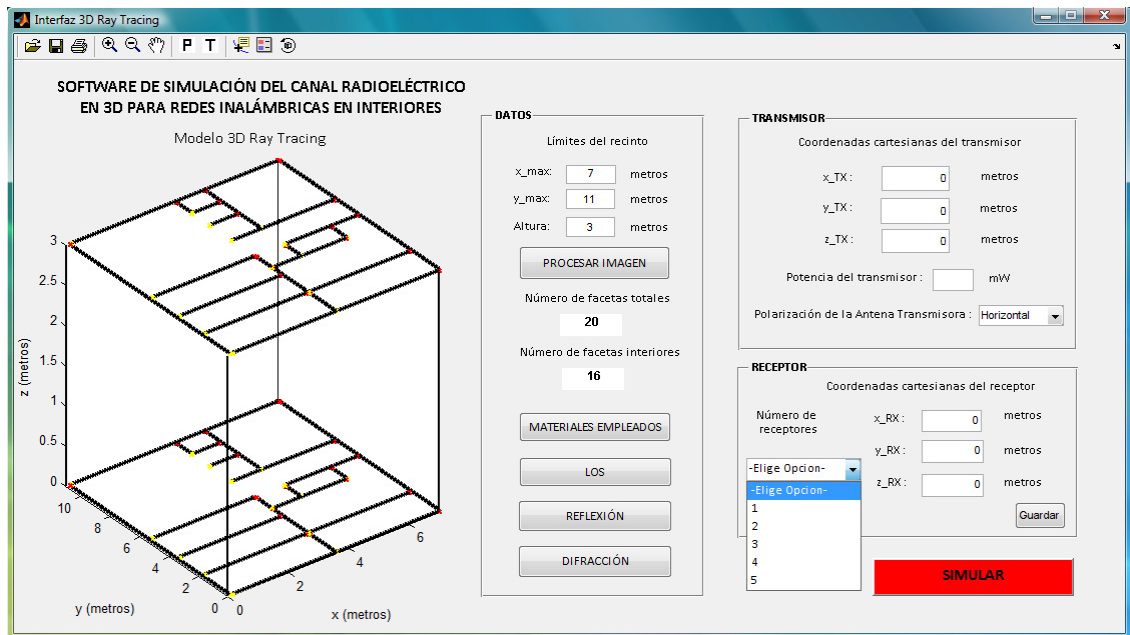


Figura 7.11: Interfaz 3D después de procesar plano

7.3.2.2.- Resolución

El siguiente paso es indicar la resolución con la que se van a calcular y mostrar los resultados. Esto se hará para la interfaz 2D y para la interfaz del método cost.

La resolución hace referencia al grado de exactitud con el que el programa presentará los resultados, es decir, la separación en metros entre los receptores virtuales que mostrarán la cobertura de la zona deseada. Cuanto menor sea la resolución, mayor será la precisión de los resultados a costa de un aumento de la carga computacional.

Se podrá elegir entre tres valores para este parámetro:

- Resolución '*Alta*' (0.02 metros) implicará una mayor calidad y precisión de los resultados, a costa de una mayor carga computacional.
- Resolución '*Media*' (0.05 metros): busca un equilibrio entre la calidad de la imagen y el tiempo de simulación.
- Resolución '*Baja*' (0.1 metros): es la que requiere menor tiempo de simulación. Como contrapartida, la calidad de la imagen no es muy óptima.

Como es de esperar una mayor resolución requiere tiempos de cómputo en las simulaciones, más altos. Lo ideal a la hora de realizar un despliegue es empezar con una resolución baja e ir aumentándola a medida que se tiene más clara la solución final.

Lo siguiente será ir pulsando sobre los botones de '**MATERIALES EMPLEADOS**', '**LOS**', '**REFLEXIÓN**' y '**DIFRACCIÓN**' y completar los datos que se piden.

7.3.2.3.- Materiales Empleados

Los diferentes materiales que constituyen los obstáculos presentes en el ambiente de estudio se van a caracterizar mediante sus propiedades eléctricas (μ , σ y ϵ).

En primer lugar, la interfaz muestra una ventana donde se informa de los materiales que se utilizan por defecto en el simulador, con sus respectivos valores de permitividad compleja y espesor. Esta ventana (Figura 7.12) es meramente informativa.

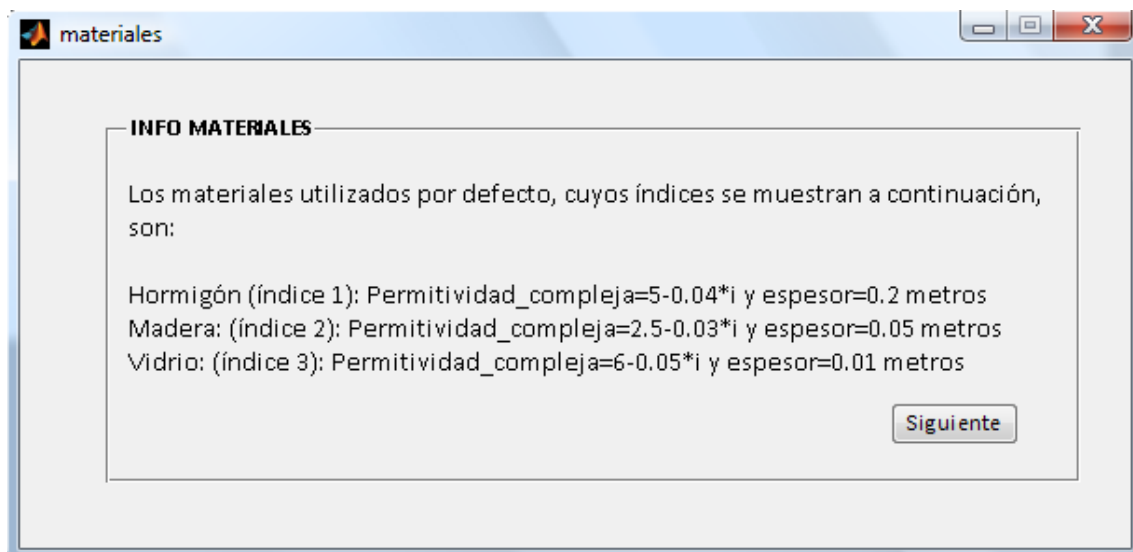


Figura 7.12: Pantalla de información de materiales por defecto

Al pulsar sobre el botón 'Siguiete', la interfaz muestra otra ventana con un cuadro de diálogo que pregunta si se quiere definir algún otro material que no esté entre los anteriores.

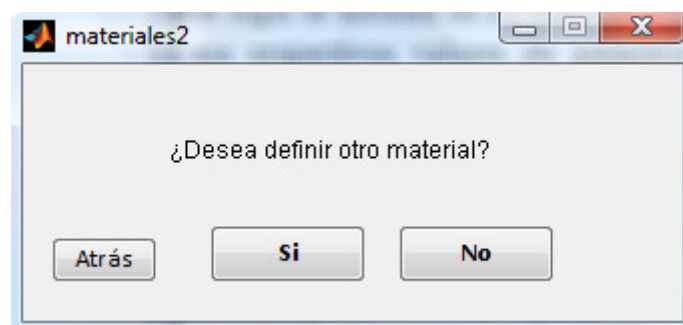


Figura 7.13: Cuadro de diálogo empleado en preguntas

Al hacer clic sobre el botón 'Si', aparecerá otra ventana para introducir la permitividad, el espesor y el nombre correspondiente al material que se quiere emplear para una potencia de 2.4 GHz.

The image shows a software window titled "materiales3" with a standard Windows-style title bar. Inside the window, there is a form titled "DEFINIR MATERIAL". The form contains three input fields: "Permitividad compleja" with a value of "0" and the unit "F/m", "Espesor (en metros)" with a value of "0" and the unit "metros", and "Nombre" which is currently empty. At the bottom of the form, there are two buttons: "Atras" and "Guardar".

Figura 7.14: Formulario para introducir nuevos materiales

Una vez se ha rellenado todos los datos del formulario, se guarda la nueva información y se vuelve a la segunda ventana donde la interfaz volverá a preguntar si se quiere seguir introduciendo nuevos materiales.

Cuando se indica que no se quieren introducir más materiales, la interfaz pasa a nuevo estado (Figura 7.15) en el que se definirá de qué está compuesto cada una de las distintas facetas.

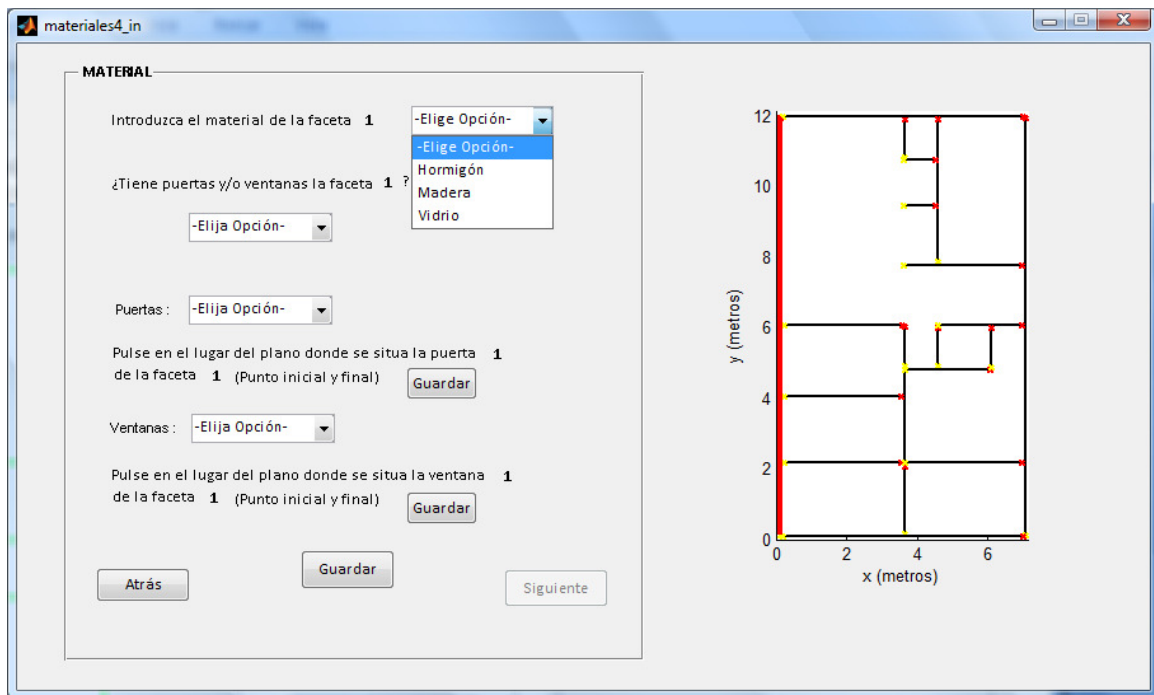


Figura 7.15: Interfaz para asignar el material a cada faceta

Para cada faceta (que se marca con una línea roja), se debe elegir, en primer lugar, el tipo de material que la compone y a continuación indicar si ésta tiene puertas o ventanas. Si se ha elegido esta última opción, se deberá hacer clic con el ratón sobre el plano para marcar el punto inicial y final de esa puerta o ventana en la línea de la faceta correspondiente.

Después de haber hecho esto para una pared determinada, haremos clic en ‘Guardar’ y se marcará en rojo la siguiente, y así sucesivamente. Se puede observar como en la figura 7.15 el botón ‘Siguiente’ no está activo de forma que hasta que no se hayan introducido correctamente todos los datos necesarios no se pase a la siguiente ventana, ya que esto produciría un error en el momento de la simulación.

Para habilitar y deshabilitar botones o modificar cualquier propiedad de los elementos de la interfaz empleamos las funciones *get* y *set*. Para deshabilitar el botón de Siguiente se ha hecho así:

```
>> set(handles.pushbutton5, 'Enable', 'Off')
```

Con esto se modifica la propiedad ‘Enable’ del elemento *pushbutton* a *Off*. En el momento que se quiera habilitar se procede de igual forma pero asignándole el valor *On*.

Cuando se haya asignado un material para cada faceta y seleccionado las posiciones de las puertas y ventanas si las hubiera, se hace clic en ‘Siguiente’.

7.3.2.4.- LOS

Cuando pulsamos sobre el botón *LOS* aparecerá la siguiente ventana.

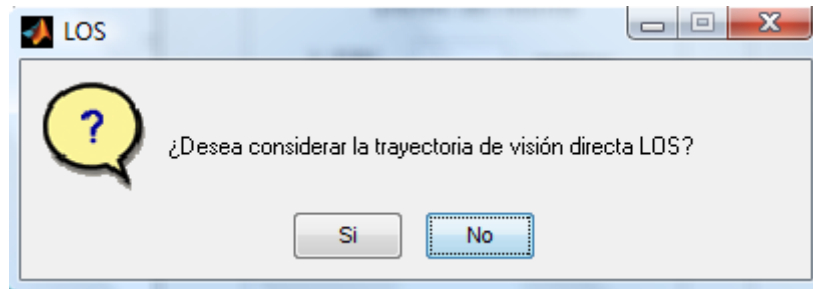


Figura 7.16: Pregunta para elegir si considerar *LOS* o no

Aquí solo se debe elegir si entre el Transmisor y el Receptor hay línea de visión directa o no.

Si hay visión directa entre los equipos se realiza el cálculo de campo recibido, teniendo únicamente en cuenta la atenuación por la distancia entre ambos; si no fuera el caso, a la atenuación se le añadiría la experimentada por la transmisión a través de las paredes, por lo que el cálculo del rayo directo se realiza aunque no haya *LOS*.

7.3.2.5.- Reflexión

El cálculo de los rayos reflejados se cimienta en la ley de reflexión de *Snell*, teniendo en cuenta la aparición de un coeficiente de reflexión. Así pues el ángulo de salida del reflejado es el mismo que el incidente en el plano definido por el rayo incidente, la normal a la superficie sobre la que incide y el rayo reflejado.

Cuando se hace clic sobre el botón 'REFLEXIÓN' aparecerá en primer lugar el siguiente cuadro de diálogo:

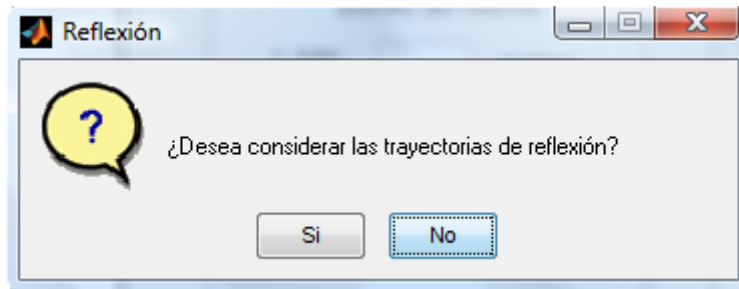


Figura 7.17: Pregunta para elegir si considerar Reflexión o no

Si se elige la opción 'Si', el siguiente paso será indicar el orden de reflexiones que tendrá lugar durante la simulación (Figura 7.18).

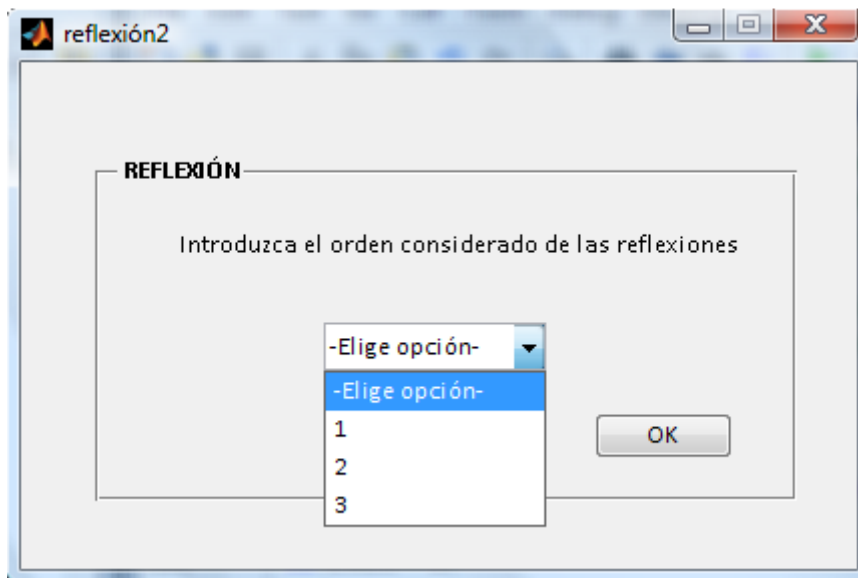


Figura 7.18: Interfaz con *popupmenu* para elegir el número de reflexiones

7.3.2.6.- Difracción

El fenómeno de la difracción se produce cuando una onda incide sobre una discontinuidad del entorno como lo puede ser la arista de una esquina o uno de los lados del marco de una ventana...etc. La arista actúa entonces como un emisor de ondas secundario, según el principio de *Huygens*, de forma que radia parte de la energía de la onda incidente hacia zonas que inicialmente serían de sombra. A frecuencias altas la

difracción depende de la geometría de la discontinuidad y de las características de la onda electromagnética que incide sobre ella (amplitud, fase y polarización).

En este apartado se contemplan los rayos difractados que proceden directamente de la fuente. La única trayectoria del rayo, antes de difractarse, es el camino directo desde transmisor hacia el receptor con visión directa, por lo que tampoco se contempla que el rayo atraviese ningún obstáculo antes de llegar al punto de difracción.

Es importante recordar que para la resolución del problema de la difracción se ha hecho uso de la Teoría Uniforme de la Difracción (*UTD* o *Uniform Theory of Diffraction*).

Al igual que para la reflexión, al pulsar en el botón ‘DIFRACCIÓN’ aparecerá una ventana para preguntar si se desea tener en cuenta las trayectorias de difracción.

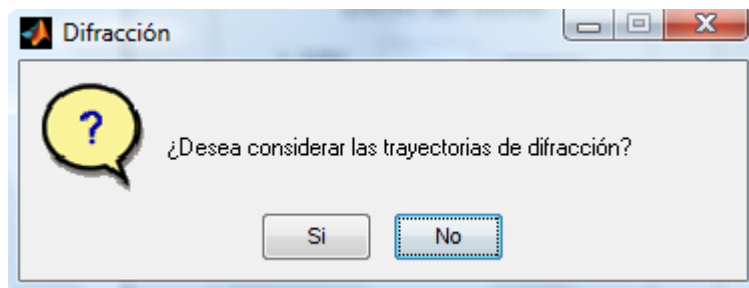


Figura 7.19: Pregunta para elegir si considerar Difracción o no

Al elegir que ‘Si’ la interfaz mostrará:

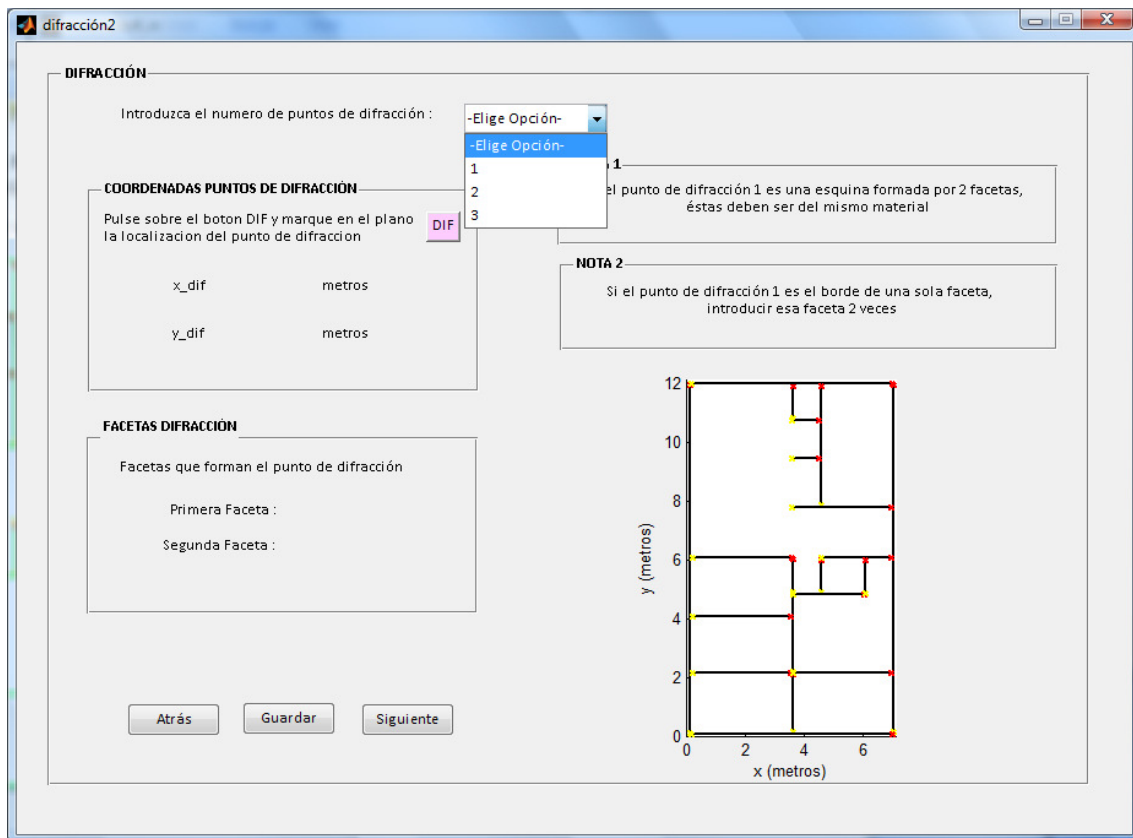


Figura 7.20: Interfaz para elegir los puntos de difracción y su localización

En primer lugar se elige el número de puntos de difracción que queremos tener en cuenta en la simulación. La carga computacional, y por consiguiente el tiempo de simulación, se incrementará con cada punto de difracción que se considere.

Después de elegir el número de puntos se hace clic sobre el botón 'DIF' y justo a continuación se procede a elegir en el plano los puntos donde se quiere tener en cuenta la difracción. A continuación se guarda la información y pulsamos Siguiete.

La siguiente ventana que aparece es la mostrada en la figura 7.21, en la que el usuario podrá elegir entre varias opciones:

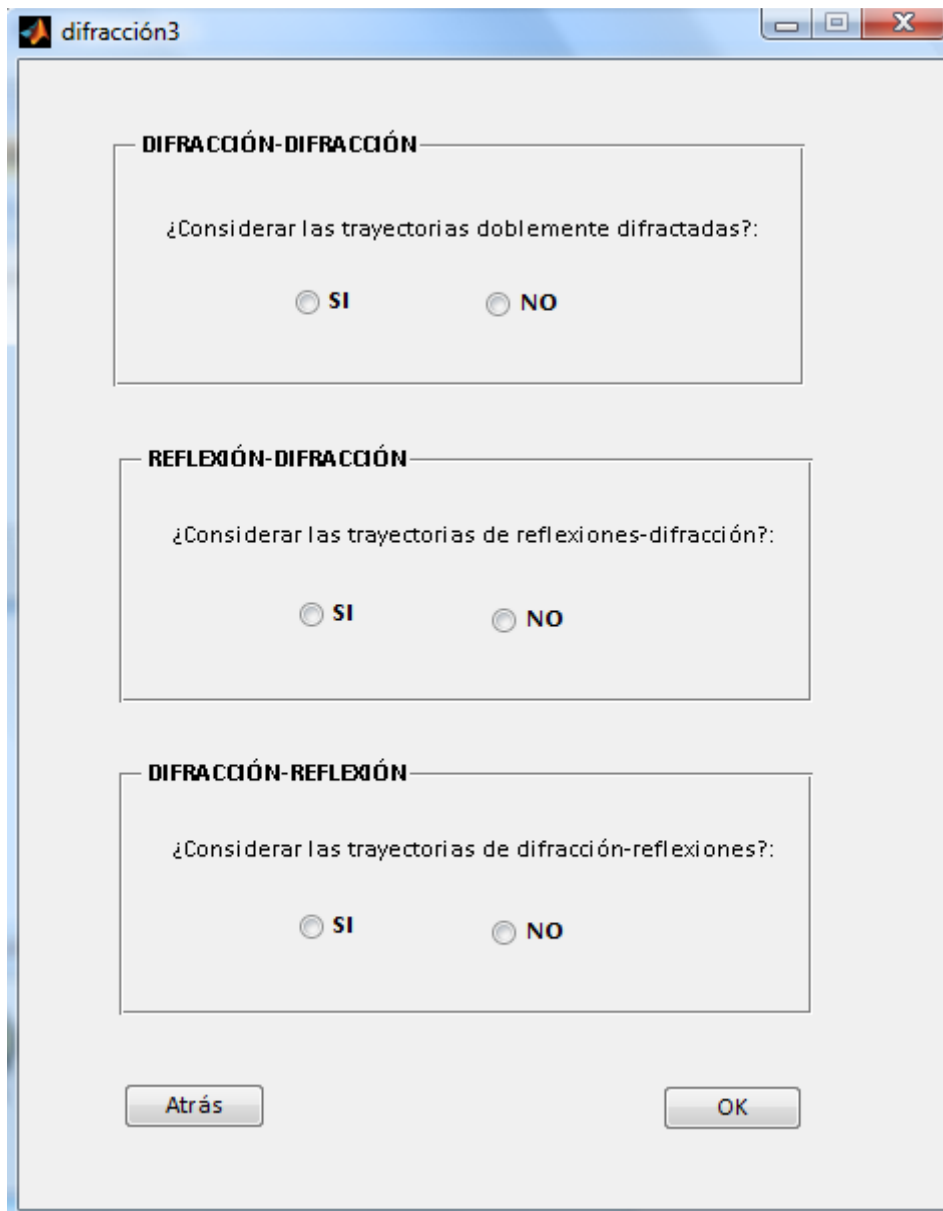


Figura 7.21: Interfaz para elegir trayectorias de difracción

Rayos doblemente difractados

El caso se trata como el rayo difractado simple, solo que ahora se considera como punto transmisor otro punto de difracción

Rayos difractados procedentes de reflexiones

En esencia el problema de estos rayos es el mismo que cuando se trataban los rayos reflejados. La única diferencia es que ahora el receptor de los rayos es el punto de difracción y no el RX. Solamente se considera una única difracción de los rayos después

de ser reflejados, ya que para difracciones de segundo orden y superiores, el campo creado es inapreciable.

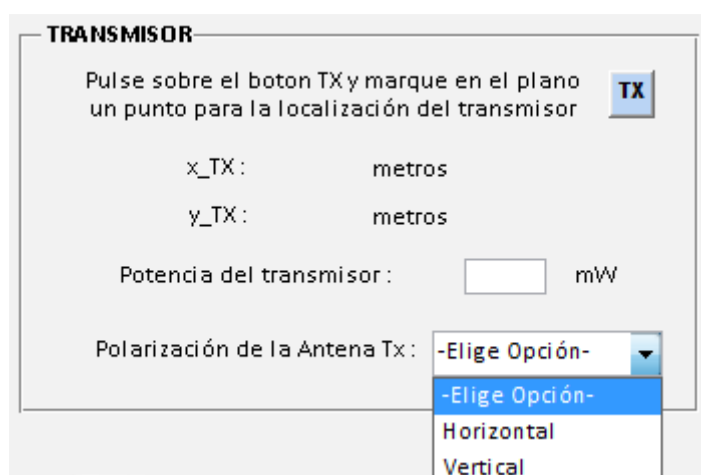
Rayos reflejados procedentes de la difracción

Tiene en cuenta las trayectorias en las que se produce una difracción y a continuación reflexión. Los rayos procedentes de la reflexión son los que llegan al receptor.

7.3.2.7.- Transmisor

Se ubicará el transmisor en el punto deseado por el usuario mediante un clic sencillo con el ratón sobre el área de trabajo. Estas coordenadas se cargan en una variable, que permitirá más adelante realizar los cálculos de distancia. El mismo procedimiento se aplica para ubicar el receptor.

Destacar que se ha limitado el área de trabajo en la interfaz al área de la imagen cargada por el usuario. Si el usuario ubica el transmisor o el receptor fuera del área de trabajo, aparece un mensaje de error.



El formulario, titulado "TRANSMISOR", contiene las siguientes opciones:

- Una instrucción: "Pulse sobre el boton TX y marque en el plano un punto para la localización del transmisor" acompañada de un botón "TX".
- Campos de entrada para las coordenadas: "x_TX : metros" y "y_TX : metros".
- Un campo de entrada para la potencia: "Potencia del transmisor : [] mW".
- Un menú desplegable para la polarización: "Polarización de la Antena Tx: -Elige Opción-", con opciones "Horizontal" y "Vertical".

Figura 7.22: Formulario para introducir datos del transmisor

Se elegirá un valor para la potencia, que como máximo será de 100mW. Un valor aconsejable será de 50 mW.

El último dato a elegir en el transmisor será la polarización de la antena transmisora.

7.3.2.7.1.- Potencia de Transmisión

Los niveles de potencia transmitida en una red inalámbrica en interiores son los fijados por la norma ETSI EN 300 328, de tal manera que ésta no puede sobrepasar el valor de 100mW de Potencia Isótropa Radiada Equivalente (PIRE). Las condiciones de cálculo y medida de la potencia están especificadas en dicha norma. En caso de utilizar antenas con ganancia superior a 0dBi, se controlará la potencia del transmisor de tal manera que el exceso de potencia quede compensado y no se rebase el límite establecido de 100mW.

Como ejemplo, los parámetros de potencia de transmisión disponibles para un Punto de Acceso de la Serie Cisco Aironet 1100, obtenidos de su hoja de catálogo, son:

- 100 mW (20 dBm)
- 50 mW (17 dBm)
- 30 mW (15 dBm)
- 20 mW (13 dBm)
- 5 mW (7 dBm)
- 1 mW (0 dBm)

El parámetro de potencia máxima variará en función de las leyes de cada país. En los ejemplos de simulación realizados durante la realización de este Proyecto se ha empleado 50mW de potencia, eligiendo en algún ejemplo 100mW para observar la diferencia obtenida en el receptor.

7.3.2.7.2.- Polarización

7.3.2.7.2.1.- Polarización Horizontal

También llamada polarización *soft* o *TE*. Tiene lugar cuando el vector campo eléctrico de la onda incidente es perpendicular al plano de incidencia, definido como el

plano que contiene al rayo incidente y al reflejado (según las leyes de *Snell* para la reflexión, el rayo incidente y el reflejado están contenidos en el mismo plano) (figura 7.23).

El campo magnético asociado a la onda estará contenido en dicho plano, ya que, al considerarse campo lejano, $H \perp E$.

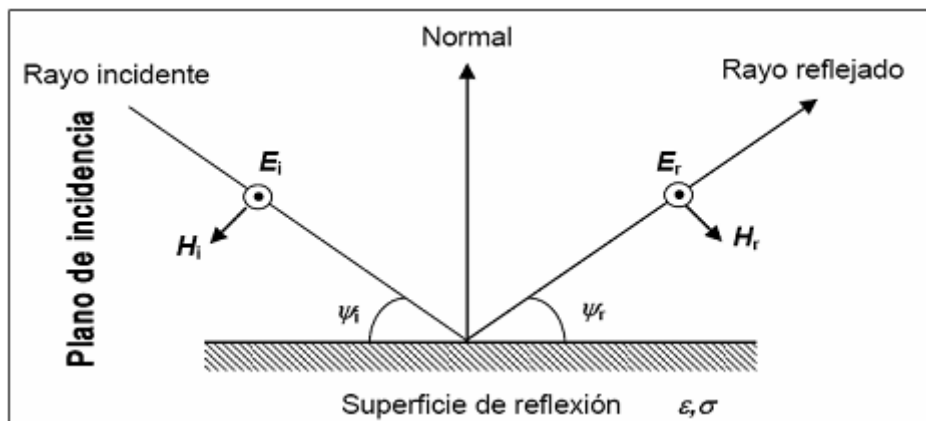


Figura 7.23: Reflexión-Polarización Horizontal

El ángulo de incidencia ψ_i coincide con el ángulo de reflexión ψ_r , según la ley de Snell ($\psi_i = \psi_r = \psi$). El coeficiente de reflexión cuando la incidencia es *soft* es:

$$R_{\perp}(\psi) = \frac{\sin\psi - \sqrt{\epsilon_r - \cos^2\psi}}{\sin\psi + \sqrt{\epsilon_r - \cos^2\psi}}$$

7.3.2.7.2.2.- Polarización Vertical

Tiene lugar cuando el vector campo eléctrico de la onda incidente está contenido en el plano de incidencia (figura 7.23). El campo magnético será, pues, perpendicular al mismo.

A esta polarización también se le denomina polarización *hard* o TM.

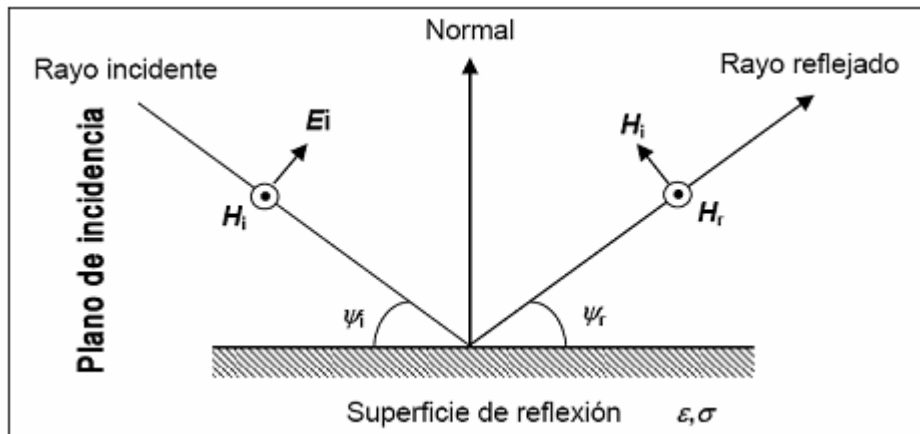


Figura 7.24: Reflexión-Polarización Vertical

El coeficiente de reflexión para una polarización vertical es:

$$R_{\perp}(\psi) = \frac{\epsilon_r \sin \psi - \sqrt{\epsilon_r - \cos^2 \psi}}{\epsilon_r \sin \psi + \sqrt{\epsilon_r - \cos^2 \psi}}$$

7.3.2.8.- Receptor

Para ubicar el receptor pulsamos sobre el botón RX y seguidamente hacemos clic sobre el plano en el punto donde queramos tener el receptor.

RECEPTOR

Pulse sobre el boton RX y marque en el plano un punto para la localización del receptor **RX**

x_RX: metros

y_RX: metros

Figura 7.25: Formulario para ubicar receptor

7.3.2.9.- Calcular potencia


Una vez cargado el plano, los datos de transmisión, y ubicados el transmisor y el receptor, se procede a calcular la potencia recibida en el punto indicado por el usuario. Para ello pulsamos en el botón ‘SIMULAR’. La potencia se calcula según el algoritmo de cada modelo.

7.3.2.10.- Cobertura Radioeléctrica

La simulación del área de cobertura se realiza calculando la potencia recibida, como se describió anteriormente, en cada uno de los puntos del área de estudio. El valor de potencia estimado se asocia a un color que se despliega sobre el punto en cuestión. La proporción de dichos colores con la potencia recibida en dBm, se muestra en la barra de color (*colorbar*).

La cobertura implica tener un nivel de señal en las zonas de interés superior a los -80dBm. Así los usuarios podrán tener acceso a determinados servicios y privilegios de red.

7.3.2.11.- Ayuda

Si tenemos alguna duda de los pasos a seguir para el correcto funcionamiento de la herramienta se puede pulsar sobre el icono  que aparecerá en la ventana principal de la interfaz y que nos mostrará la siguiente información:

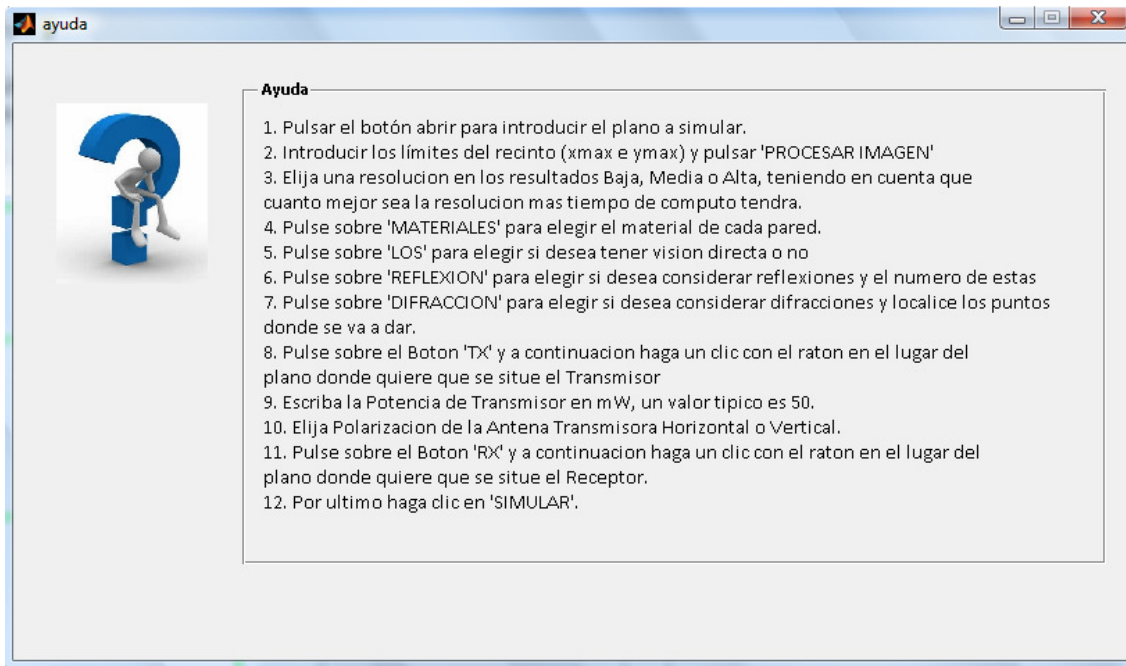


Figura 7.26: Interfaz de ayuda

7.3.2.12.- Barra de progreso

Mientras se ejecuta un archivo o una función en Matlab, el usuario no tiene idea de cuánto tiempo falta para completar la simulación. Implementando una barra de progreso, el usuario podrá conocer el estado de la simulación así como del tiempo que falta para completarse. Es posible utilizar en Matlab el comando *waitbar*, pero este no es tan flexible como la típica barra de progreso por tres razones:

- No muestra el porcentaje completado.
- No muestra el tiempo restante para finalizar la ejecución.
- Si el usuario cierra la barra de progreso antes de que se haya completado, esto generará un error.

Wait Bar Matlab:

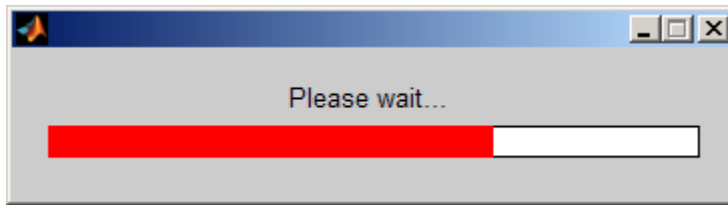


Figura 7.27: Barra de progreso original de Matlab

Wait Bar Modificada:

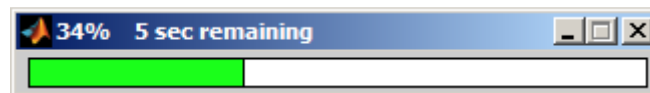


Figura 7.28: Barra de progreso modificada

Se utiliza para ello el fichero *progressbar.m*. Este fichero toma dos parámetros: el primero es el tiempo de la tarea que se ha completado, por lo que la entrada variará entre 0 y 1; el segundo parámetro es la posición.

La posición determinará dónde queremos que aparezca la barra de progreso en la pantalla. La posición de la barra de progreso vendrá especificado por:

[x, y] – Posición de la esquina inferior izquierda en unidades normalizadas (0.0 - 1.0)

- 0 – En el centro (por defecto)
- 1 – Superior derecha
- 2 – Superior izquierda
- 3 – Inferior izquierda
- 4 – Inferior derecha
- 5 - Posiciones [x, y] aleatorias

El parámetro de salida se utiliza, principalmente, para salir de la ejecución. Si la barra de progreso está activa en la pantalla, el parámetro de salida valdrá 0. Cuando el usuario cierra la barra de progreso, la función devolverá un 1.

Este, es un rápido ejemplo de cómo usar la barra de progreso en nuestro código:

```
clear all
for x=1:100000
stopBar= progressbar(x/100000,0);
if (stopBar) break; end
end
```

Esto es lo que podremos ver mientras se ejecuta el programa:

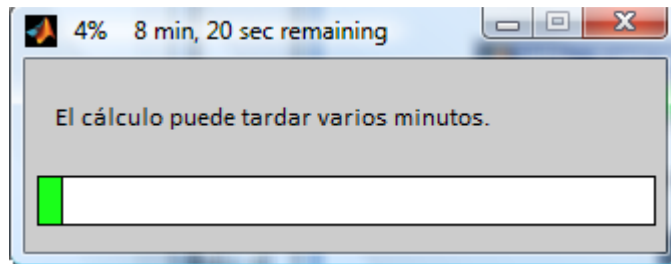


Figura 7.29: Barra de progreso del simulador *ESIndoor*

Si se intenta cerrar la barra de progreso antes de llegar al 100% recibiremos el siguiente cuadro de diálogo.



Figura 7.30: Cuadro de diálogo al cerrar barra de progreso

7.3.2.13.- Excepciones de la Interfaz

Se emplearan cuadros de mensajes para indicarle al usuario que existe algún error, se quiera mostrar información o hacer una pregunta cuya respuesta sea sí o no. Existen algunos tipos en Matlab que directamente muestran el cuadro de diálogo aunque también se pueden personalizar.

Las siguientes sentencias se ubican debajo de la función correspondiente para mostrar el cuadro de mensajes predefinido en Matlab:

```
warndlg('Esto es un aviso', 'AVISO');
```

```
errordlg('Esto es un mensaje de error', 'ERROR');
```

```

helpdlg('Esto es una ayuda', ' AYUDA ');
msgbox('Esto es un cuadro de mensaje', ' MENSAJE ');
questdlg('Esto es una pregunta', ' PREGUNTA ');
inputdlg('Esto es un dato', 'DATO');

```

Parte de nuestro m-file queda de la siguiente forma cuando ocurre una excepción, pulsamos sobre Ayuda o hacemos una pregunta:

```

% Se ejecuta cuando se producen errores en la inserción de datos en la interfaz.
function Error_c_Callback (hObject, eventdata, handles)
errorDlg ('Error: Los datos introducidos no son correctos', 'Excepciones');

```

```

% --- Se ejecuta cuando se pulsa sobre el botón de Ayuda.
function Ayuda_Callback (hObject, eventdata, handles)
helpdlg ('Esto es una ayuda', 'Ayuda ');

```

```

% --- Se ejecuta cuando para realizar preguntas
function Pregunta_Callback (hObject, eventdata, handles)
questdlg ('¿Quieres definir otro material?', 'Preguntas ');

```

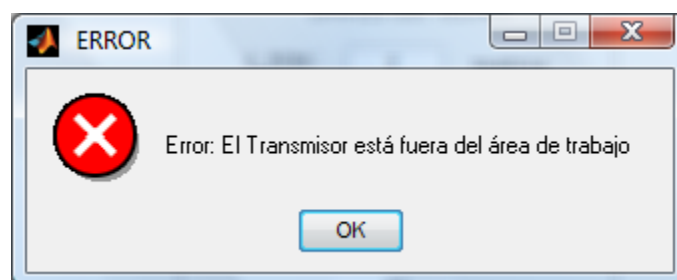


Figura 7.31: Cuadro de diálogo para indicar un error en la adquisición de datos

7.3.3.- Crear ejecutable de la herramienta en Matlab

Matlab posee un compilador que convierte nuestros programas en archivos .exe que pueden ejecutarse sin necesidad de abrir Matlab o en otros ordenadores que no tengan instalado este software. Para ejecutar este compilador, es necesario tener instalado Simulink.

Para ello Matlab posee una herramienta que se ejecuta desde línea de comandos:

```
>> deploytool
```

Este comando abre la siguiente ventana con los pasos para hacer un ejecutable:

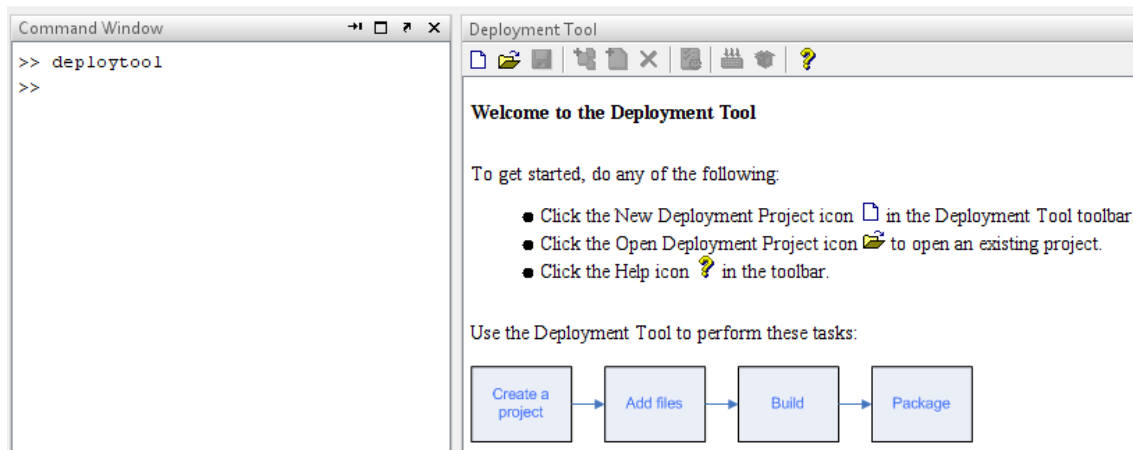



Figura 7.32: Ventana de inicio para crear ejecutable Matlab

En primer lugar, se hace clic sobre el botón de nuevo  y se abrirá el siguiente cuadro de dialogo:

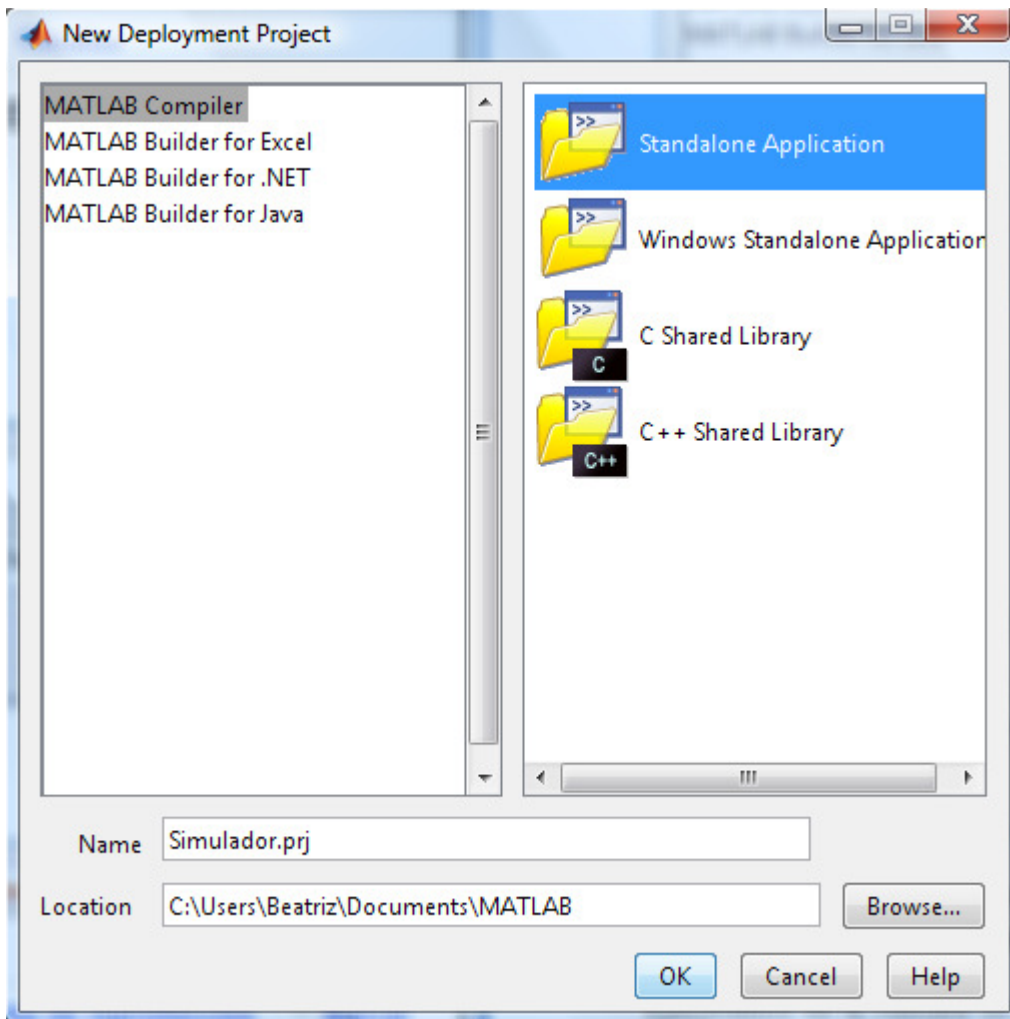



Figura 7.33: Ventana para indicar nombre del proyecto

Se dejan marcadas las opciones que aparecen por defecto y se escribe un nombre para el proyecto, en este caso va a ser 'simulador', y se presiona sobre OK.

El siguiente paso es marcar la carpeta '*Main function*' y hacer clic sobre el icono añadir fichero  (Fig.7.33).

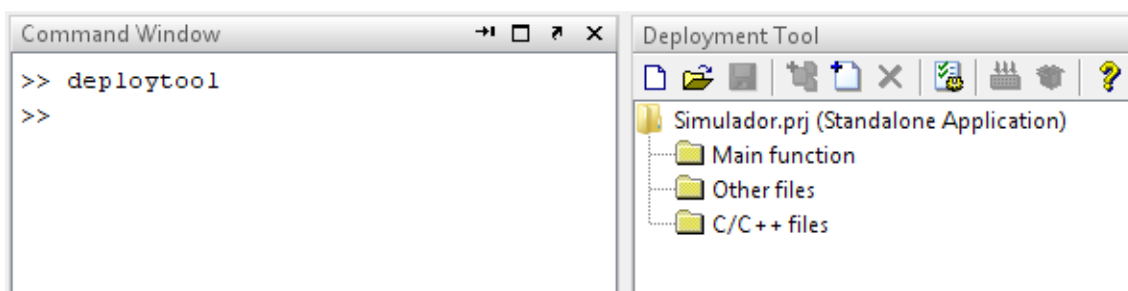


Figura 7.34: Ventana para añadir ficheros al proyecto

Se elige el fichero .m principal del proyecto, en este caso 'portada.m' que será el primero que se ejecutará al abrir el .exe.

A continuación, se hará lo mismo añadiendo los demás archivos que componen el proyecto en 'Other files' quedando algo como la figura 7.35:

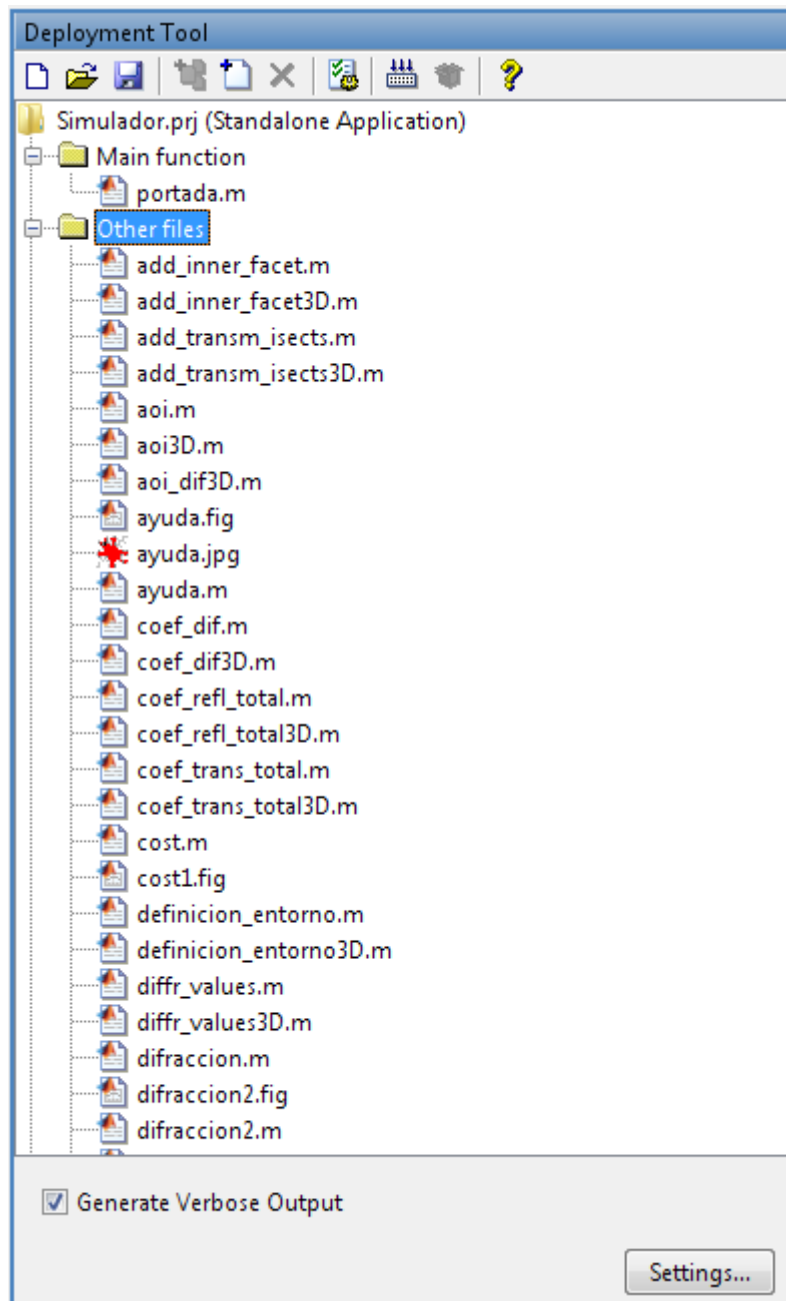



Figura 7.35: Ficheros principal y otros añadidos al proyecto

A continuación se pulsa sobre el icono  'Open Project settings window' y nos aparece otra ventana como la siguiente:

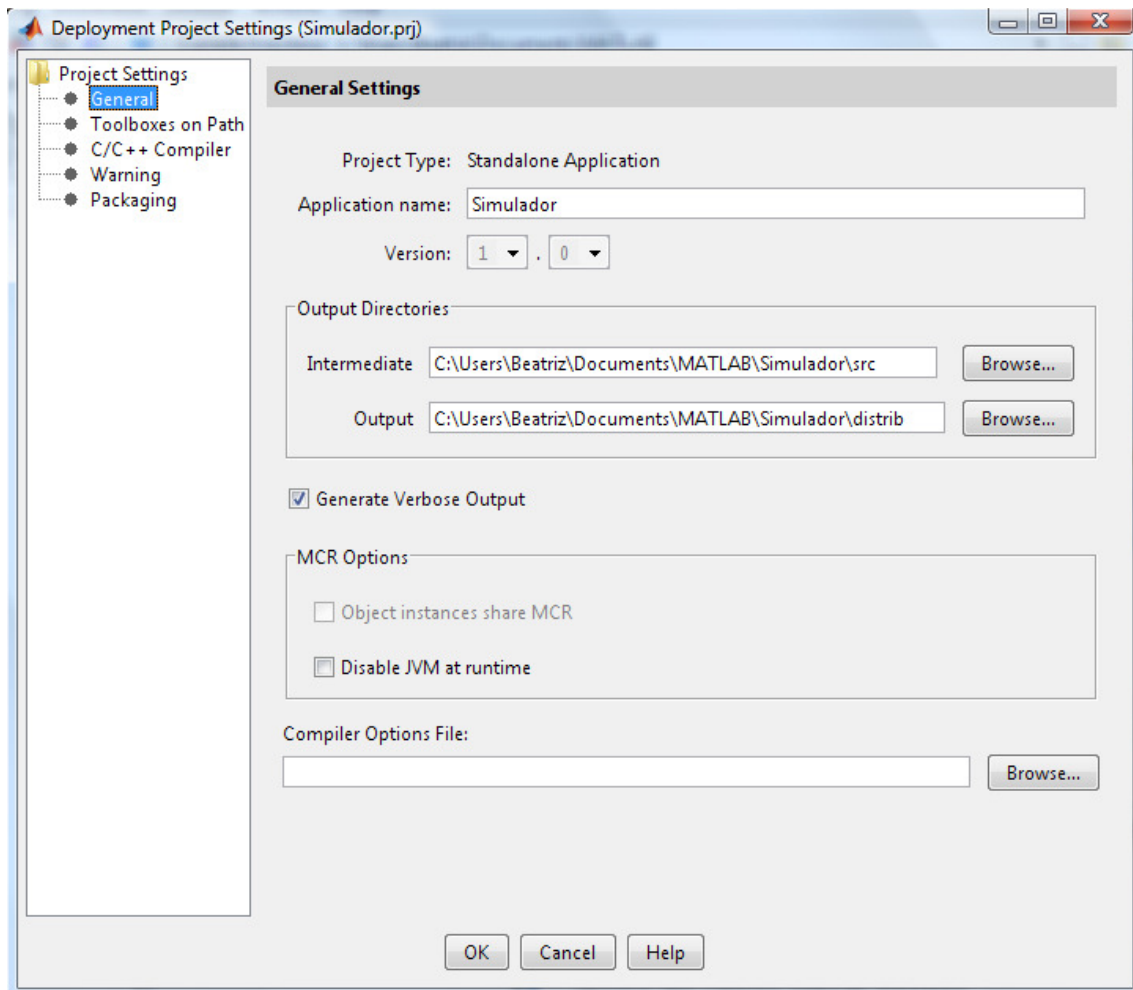


Figura 7.36: Ventana de opciones

El siguiente paso es seleccionar ‘*Packaging*’ y marcar el *checkbox* ‘*Include Matlab Component Runtime (MCR)*’. Esta opción es muy importante ya que al activar esa casilla se empaquetará también el *Runtime* para poder ejecutar el programa sin necesidad de tener instalado Matlab.

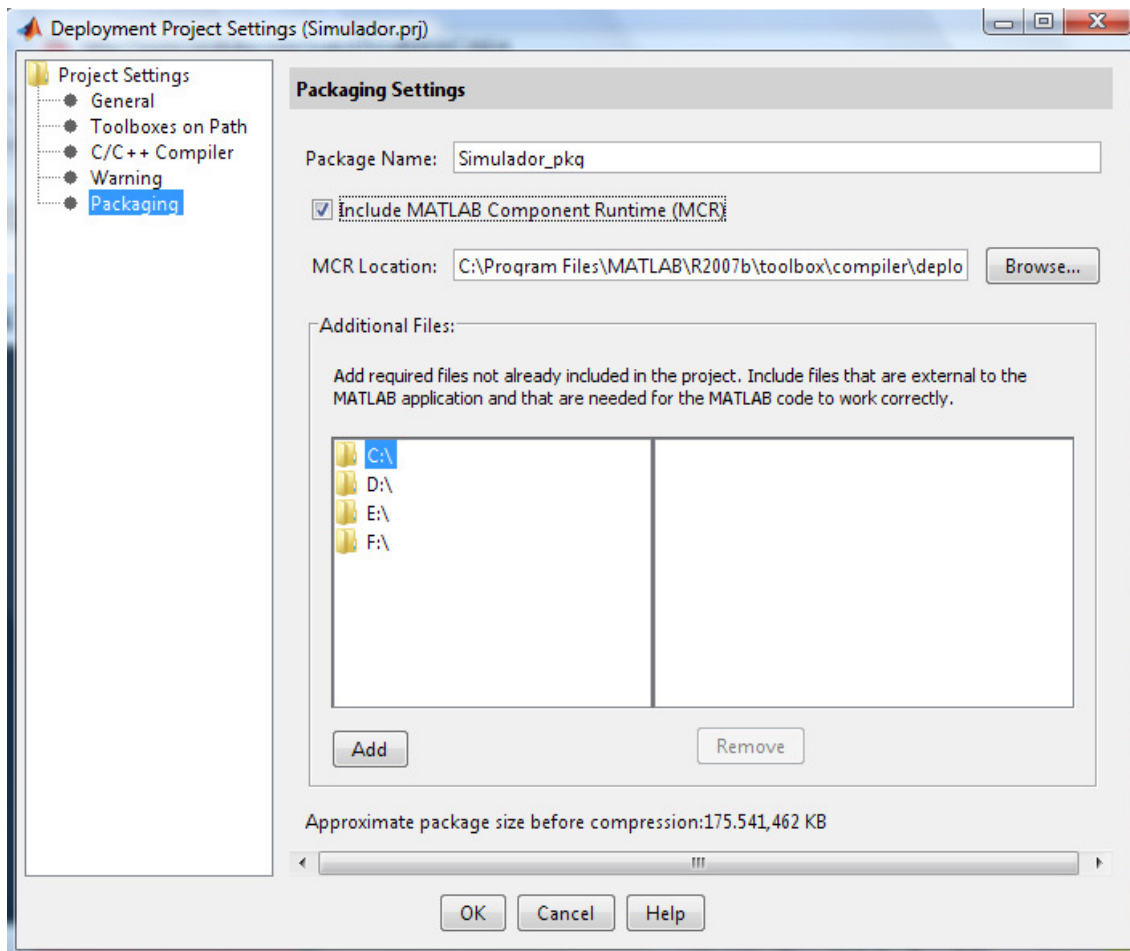



Figura 7.37: Incluir el *Runtime* de Matlab al ejecutable

Se pulsa *OK* y por último se hace clic en el icono  '*Built the project*'.

Una nueva ventana en la parte inferior de la pantalla principal de Matlab indicará que el ejecutable se está construyendo (Figura 7.38)

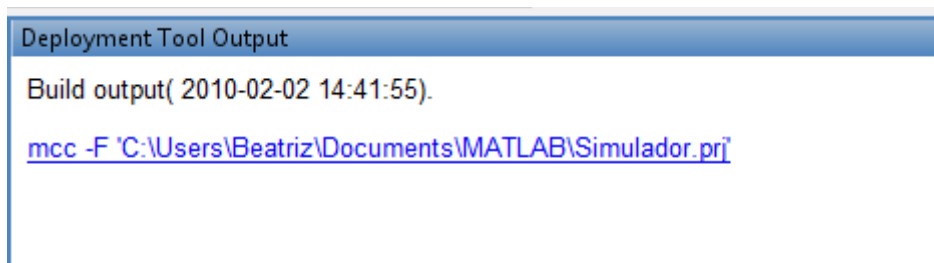


Figura 7.38: Construyendo el fichero .exe

Finalmente, si la compilación se ha completado sin errores aparecerá el siguiente mensaje:

```
Deployment Tool Output
Build output( 2010-02-02 14:41:55).
mcc -F 'C:\Users\Beatriz\Documents\MATLAB\Simulador.prj'

Compiler version: 4.7 (R2007b)
Warning: Duplicate directory name: C:\Users\Beatriz\Documents\MATLAB.
mcc -o 'Simulador' -W 'main' -d 'C:\Users\Beatriz\Documents\MATLAB\Simulador\src'
Processing C:\Program Files\MATLAB\R2007b\toolbox\matlab\icc\icc.mcc
Processing C:\Users\Beatriz\Documents\PFC-22-1-10\ayuda.fig
Processing C:\Users\Beatriz\Documents\PFC-22-1-10\cost1.fig
Processing C:\Users\Beatriz\Documents\PFC-22-1-10\difraccion2.fig
Processing C:\Users\Beatriz\Documents\PFC-22-1-10\difraccion2_3D.fig
Processing C:\Users\Beatriz\Documents\PFC-22-1-10\difraccion3.fig
Processing C:\Users\Beatriz\Documents\PFC-22-1-10\interfaz2D.fig
Processing C:\Program Files\MATLAB\R2007b\toolbox\images\icc\icc.mcc
Processing C:\Users\Beatriz\Documents\PFC-22-1-10\interfaz3D.fig

< [ ]
Compilation completed
```

Figura 7.39: Programa ejecutable creado